Joni Aalto

Opas projektien pariin digitaalisessa maailmassa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikan koulutusohjelma

Insinöörityö

8.5.2017

Metropolia

Tekijä Otsikko Sivumäärä Aika	Joni Aalto Opas projektien pariin digitaalisessa maailmassa 31 sivua 8.5.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Ohjaaja	Yliopettaja Markku Karhu

Insinöörityön aiheena oli projektityö digitaalisessa maailmassa. Sen tavoitteena oli luoda opas projektien parissa työskentelyyn ohjelmistotuotannossa ja selkeyttää ajatusta projektinhallinnasta.

Insinöörityössä perehdyttiin ohjelmistojen kehittämisen projektinhallintaan yleisesti ottaen sekä pohdittiin muutamia projektinhallintamalleja. Käsittelyssä olivat Scrum-malli, Kanban-menetelmä, vesiputousmalli, RUP-malli ja Extreme Programming -metodi. Työssä etsittiin yhtäläisyyksiä näistä malleista sekä perusteltiin projektityömallin valintaa. Työssä haettiin tietoa monipuolisesti eri verkkosivustoja ja kirjallisuutta käyttäen sekä tekijän omasta kokemuksesta IT-alalla.

Insinöörityössä selvisi, että jokainen projektityömalli liittyy toisiinsa. Projektityömalleista löytyi paljon yhtäläisyyksiä. Yhtäläisyydet olivat tavoissa toteuttaa projektia, vaikka malli tai metodi oli kovin erilainen muilta piirteiltään. Yrityksen sisällä voi olla useampaa eri mallia käytössä riippuen asiakkuudesta ja projektista.

Insinöörityöraportti on tarkoitettu alaa opiskeleville ja projektitöiden parissa työskenteleville henkilöille. Jokaisen ohjelmisto- tai digitaalisen toimiston henkilöstöön kuuluvan, jotka usein projektien kanssa tekevät töitä, tulisi tutustua eri projektimalleihin. Siitä saa parhaan käsityksen käytetystä projektimallista, ja mahdollisesti selviävät myös syyt kyseisen mallin tai mallien valintaan eri projekteissa.

Avainsanat	projektinhallinta, projektityömallit

Author	Joni Aalto
Title	Guide to projects in digital world
Number of pages	31 pages
Date	8.5.2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Instructor	Markku Karhu, Principal Lecturer

In this thesis different principles of project management in software production were studied and different common project models were compared. Project models such as Scrum, Kanban, Waterfall, RUP and Extreme Programming were taken into account. The main reason for this thesis was to create a guide on good basic knowledge of project management in digital world.

In the study it was found that every project model is very similar to each other. The differences between models concerned more the terms used and some working practises. The most interesting results were based on the comparison.

The result of this thesis gives a good basic view on project management and project models for students who want to study software engineering and for people who already work in the field. It is believed that all colleagues who work in the field should know the differences of the software project models and be capable of judging the reasons why a project model was chosen to be followed in a particular project.

In this thesis, the data and information were collected data from literature and from various internet sources. The author also has personal experience of working in different IT projects for many years.

Keywords	project management, project models

Sisällys

1 Johdanto

Olen itse hiljattain vaihtanut työpaikkaani teknisemmästä tekemisestä projektinhallintaan digitaalisessa toimistossa. Tästä inspiroituneena olen kiinnostunut eri projektinhallintamalleista, joita voin käyttää oman työn suunnittelemisen mallina, jotta tulevaisuuden projekteissa projektinhallintamalli olisi juuri oikeanlainen oikeaan projektiin.

Olen työurani aikana toiminut kahdeksan vuotta teknisen asiantuntijan tehtävissä, josta siirryin projektipäällikön tehtäviin, joissa olen toiminut nyt noin vuoden verran. Projektinhallinta tuli viimeisinä vuosina asiantuntijana tutuksi, kun minulle annettiin mahdollisuus ottaa vastuu digitaalisten palveluiden suunnittelusta, kehityksestä ja hallinnasta.

Nykyisessä työpaikassani käytetyimmät projektinhallintamallit ovat Scrum-malli, Kanban ja vesiputousmalli. Näistä itselläni eniten kokemusta on Scrumin ja Kanbanin yhdistelmästä eli niin sanotusta Scrumbanista.

Tämän insinöörityön pääasiallinen tavoite on selvittää erilaisia ohjelmistotuotannon projektien hallinnan malleja ja niiden ominaisuuksia sekä laatia raportti, joka on eräänlainen opas projektinhallintaan ohjelmistotuotannossa. Opas toivottavasti selkeyttää ohjelmistoprojektien käsitteitä ohjelmistokehityksen parissa työskenteleville ja niille opiskelijoille, joilla on mielenkiintoa ohjelmistotuotantoa kohtaan.

2 Ohjelmistoprojektien hallinta

Voisi sanoa, että jokaisessa työpaikassa on jonkinasteista projektityöskentelyä jossakin vaiheessa tuotekehityksen elinkaarta; niiden muodot ja olemukset vain eroavat usein toisistaan. Samaa kaikissa projekteissa on kuitenkin se, että jokaiselle projektille on todennäköisesti määritelty kustannus-, aika- ja lopputavoitteet. Ilman näitä tavoitteita projektia ei hallita oikealla tavalla, ja jokainen tavoite vaikuttaa toisiinsa.

Kuvassa 1 näytetään konkreettisesti, miten projektinhallinnan tavoitteet vaikuttavat. Tämä tarkoittaa sitä, että jos jokin kuvassa 1 olevista sivuista kasvaa, se vaikuttaa välittömästi muihin sivuihin. Kun sivut pysyvät vakiona, syntyy todennäköisimmin haluttu tuote tai palvelu, jota kuvataan keskellä olevalla kolmiolla. [1.]

Kuva 1. Projektikolmio [1].

Kuva ei ole täydellinen totuus, vaan riippuu projektin olosuhteista ja tyypistä. Esimerkiksi vaikka aikataulua kiristetään, se saattaa silti maksaa enemmän tai vähemmän, mutta todennäköisesti jos sovituista tavoitteista poiketaan, on suurempi riski, että projekti ei kuitenkaan ole onnistunut. [1.]

Ohjelmistotuotantoon liittyy monia tyypillisiä projektin vaiheita:

- Esitutkimusprosessi, johon kuuluu keskeisenä osana vaatimusmäärittely. Vaatimusmääritelmä pyrkii selvittämään tuotteen tilaajan ja loppukäyttäjän asettamat vaatimukset ja toiveet. Vaatimukset ovat toiminnallisia tai eitoiminnallisia.

- Määrittelyprosessi, jonka tarkoituksena on katsoa, mitä järjestelmän tulisi tehdä. Tämä tapahtuu yleensä tutkimalla esitutkimuksesta löytyneitä asiakasvaatimuksia, jotka lopulta määrittelevät vaadittavat lopputuotteen toiminnallisuudet.

- Suunnitteluprosessi, jossa olisi hyvä erottaa toisistaan projektin johtamisen ja projektin sisällön suunnittelu. Tämä tarkoittaa, että projektille on hyvä asettaa tavoitteet, katsoa, miten ositetaan projekti, tehdään aikataulusuunnitelma, resurssisuunnitelma ja kustannussuunnitelma. Näin syntyy projektisuunnitelma.

- Toteutusprosessi, jossa käynnistetään tuotteen kehittäminen määritelmien ja valittujen toteutustapojen mukaan.

- Käyttöönottoprosessi, jossa otetaan tuote käyttöön loppukäyttäjille joko vanhan tilalle tai täysin uutena tuotteena riippuen tuotteesta. Tätä vaihetta tosin voidaan myös vaiheistaa esimerkiksi kuluttajatestauksessa.

Kun edellä olevat vaiheet on tehty, voisi luulla että tuote on valmis, mutta ohjelmistotuotannossa harvoin näin on, riippuen tuotteen luonteesta. Yleisesti tämän jälkeen puhutaan ylläpitoprojektista ja tuotetta kehitetään asiakkaan näkemysten mukaisesti. On myös mahdollista, että tuote saadaan lopullisesti valmiiksi eikä sitä enää jatkokehitetä. Yrityksessä, jossa työskentelen, onkin se ajatus, että tuotetaan vain tulevaisuuden kestäviä digitaalisia tuotteita, eli jokaisen applikaation ja ohjelmiston olisi tarkoitus olla olemassa pidempään ja kehitys olisi pysyvää. Mutta esimerkiksi peliteollisuudessa tilanne saattaa olla toinen, koska silloin tarkoituksena on luoda yksi tuote, joka ei tule olemaan pysyvä. Näin ainakin menneisyydessä ennen internetaikaa. Nykyään tämäkin näyttäisi muuttuvan, sillä pelejä kohdellaan kuin ohjelmistoja, jotka vaativat päivityksiä ja lisäsisältöä.

2.1 Riskien hallinta ja projektin onnistuminen

2.1.1 Riskien hallinta

Riskien hallinnalla tarkoitetaan prosessia, jossa tunnistetaan, analysoidaan ja vastataan todennäköisiin riskeihin projektin aikana. Riskien hallinnalla kontrolloidaan tule-

vaisuuden tapahtumia proaktiivisesti eikä reaktiivisesti. Esimerkiksi halutaan luoda täysin uudenlainen verkkokauppa, jollaista kukaan ei ole aiemmin tehnyt. Asiakas haluaisi, että tuote olisi valmis 6 kuukauden päästä, mutta kehittäjätiimin mielestä projekti vaatii ainakin 9 kuukauden työn. Jos projektipäällikkö on proaktiivinen, tiimin kanssa luodaan valmiussuunnitelma, jonka avulla ratkaistaan aikataulu-ongelma joko vähentämällä ominaisuuksia tai muuttamalla julkaisusuunnitelmaa, ja tällöin asiakas saa itse priorisoida ja kantaa vastuun. Jos projektipäällikkö on reaktiivinen, olisi luvattu, että kaikki on valmista tiettyyn päivään mennessä, mutta havahduttu vasta myöhemmässä vaiheessa, että tämä ei tule onnistumaan. Tämä tapa ajaa tiimin pahoihin ongelmiin, ja tässä menetetään sekä rahaa että aikaa. [2.]

Tapaus-esimerkkinä, jossa todennäköisesti on epäonnistuttu riskien kartoittamisessa, on Tieto-ohjelmistotalon toteuttama tuomioistuimelle tehty IT-järjestelmä RITU, jossa Niclas Storåsin analyysin mukaan Tieto joutui muuttamaan organisaatiotaan ja kärsi sisäisistä ongelmista hankkeen epäonnistumisen seurauksena. Järjestelmä oli lopulta 1,5 vuotta myöhässä, ja tästä syystä kustannukset nousivat noin 36 prosenttia suuremmiksi kuin oli suunniteltu. Konsultit huomasivat vasta tarkastuksissaan useita osa-alueita, joissa oli puutteita. Näitä puutteita ei Storåsin analyysissä sen tarkemmin avattu, mutta nämä puutteet vaikuttivat aikatauluun ja hankkeen lopputulokseen. Hanke kriisiytyi vuoden 2011 alkupuolella ja keskeytyi tilapäisesti. Hankkeen uudelleen käynnistyksessä tehtiin myös virheitä, ja määrittelyt jouduttiin tekemään kahteen kertaan. [3.]

2.1.2 Riskien hallinnan tarkoitus

Riskien hallinnan tarkoitus on tunnistaa, vähentää ja rajata riskit, tarjota järkevä perusta päätöksenteolle ja projektin suunnittelulle. Riskien hallinta ja arviointi on paras ase projektin katastrofeja vastaan. Arvioimalla ja tekemällä suunnitelmia parannetaan projektin onnistuvuuden mahdollisuuksia. Jatkuva riskien hallinta varmistaa, että suuriin ongelmiin vastataan aggressiivisesti mahdollisimman nopeasti koko projektin ajan. [2.]

2.1.3 Riskien hallinta käytännössä

Riskien hallintaa toteutettaessa ensin täytyisi tarkastaa mahdollisten ongelmien syyt, esimerkiksi:

- Projektin hallinnassa
    - Ylin johto ei tunnista tätä vaihetta projektissa.
    - Projekteja on samaan aikaan liian monta suhteessa resursseihin.
    - Aikataulut ovat mahdottomia.
    - Tuotteella ei ole omistajaa.
    - Ongelmat tiimin jäsenen tai jäsenien kanssa.
    - Ei ymmärretä projektipäällikön roolia.
    - Väärä projektipäällikkö vetämässä projektia.
- Ulkoiset riskit
    - Ennustamattomat
        - vandalismi, sabotointi
        - luonnon katastrofit
    - Ennustettavat
        - tilanne markkinoilla
        - sosiaaliset haasteet
        - inflaatio
        - media
    - Tekniset
        - teknologiat muuttuvat

        - suunnitteluprosessin ongelmat
    - Lakitekniset
        - lisenssiin tai tavaramerkkiin liittyvät ongelmat
        - sopimusrikkomukset
        - lainsäädäntö

Tämän jälkeen tulisi analysoida riskit, jotta ne saadaan jatkossa tunnistettua ja toteutuminen estettyä. Laatu- ja arviointityökaluja käytetään määrittämään ja priorisoimaan riskejä. Analysoinnissa tulisi tunnistaa riski. Tämä voidaan tehdä esimerkiksi aivoriihessä, jossa arvioidaan mahdolliset ongelmien syyt ja lähteet. Riskien arvioinnilla tarkoitetaan sitä, että kun mahdolliset riskit on löydetty, mietitään, mitä riskeille voidaan tehdä. Tiimin tulisi miettiä tässä vaiheessa vastaukset kysymyksiin, mikä voisi aiheuttaa riskin ja kuinka se vaikuttaa tähän projektiin. Tulisi määrittää vastaukset riskeille, eli tarvitaan toimivat toimintatavat riskien käsittelyyn. Tiimin tulisi vastata kysymyksiin, mitä voidaan tehdä, jotta vähennetään riskien toteutumisen mahdollisuutta, ja mitä tehdään, kun/jos ongelma ilmenee. Määritetään myös jatkuvuussuunnitelma eli toimintoja, miten hallita ongelmia. [2.]

Monesti tilanne kuitenkin tuntuu olevan se, ettei esitutkimukseen tai määrittelyyn suhtauduta kovin vakavasti ja osallistujien valinta perustuu asemaan yrityksessä ja henkilöresursseihin. Monesti tuntuu myös siltä, että sidosryhmiä on vaikea saada motivoitumaan projektista, kun konkreettista tuotetta ei ole vielä olemassa. Yksi projektipäällikön vastuualueista onkin pitää huolta tiimistään, pyrkiä motivoimaan sitä ja saada tiimi asettumaan yhteisen tavoitteen taakse. Projektiryhmän motivaatio tosin riippuu myös siitä, kuinka yksilöt liittyvät yhteen, sekä heidän kyvystään käsitellä projektin hyviä ja huonoja puolia. [4.]

Projektiin osallistuvien yksilöiden sitoutumisen ja motivaation täytyy olla rehellistä. Sen seurauksesta saadaan hyvä työilmapiiri ja kohonnut tuottavuus yksilöiden ja koko ryhmän osalta. Projektipäällikön tulee olla myös tietoinen henkilöiden taidoista, kokemuksesta, henkilökohtaisista asenteista, olosuhteista ja motivaatiosta. Kuvassa 2 kuvataan kolmion avulla miten projektitiimin rooliutuminen ja ryhmäytyminen tapahtuu. [4.]

Kuva 2. Projektikolmio henkilötasolla [5].

2.1.4 Projektin onnistuminen

Projektin onnistumista voidaan perinteisesti mitata kolmella mittarilla eli aikataululla, budjetilla ja laatutasolla. Tähän perustuu näkemys luokitella sisäisten ja ulkoisten suoritusvaatimusten kriteerit. Sisäisillä suoritusvaatimuksilla tarkoitetaan aikataululle, kustannuksille ja teknisille määrittelyille asetettuja vaatimuksia. Ulkoisilla suoritusvaatimuksilla taas tarkoitetaan sitä, että projekti ja siitä syntynyt tuote on asiakkaan hyväksymä ja sitä voidaan käyttää referenssinä. [6.]

Onnistuminen toisaalta voidaan jakaa myös kahteen osaan eli projektinhallinnan onnistumiseen ja projektista syntyneen tuotteen onnistumiseen. Projektinhallinnassa mittareina toimivat aikataulujen pitävyys, kustannustehokkuus ja laatutavoitteet. Tuotteen onnistumisen tärkeimpänä mittareina toimii se, että tuote vastaa projektin omistajan odotuksia ja asiakas on tyytyväinen.

Myyvän yrityksen olisikin hyvä toimia kumppanina, joka tarjoaa järkeviä projekteja, joista sekä asiakas että yritys hyötyvät konkreettisesti. Tällä tarkoitan sitä, että pitäisi

osata myös arvioida, milloin projektia ei kannata toteuttaa. Asiakasyritykselle se voi tarkoittaa hyvää lisäarvoa tuovaa tuotetta, ja kehittävälle yritykselle se voi tarkoittaa esimerkiksi omien kompetenssien kehitystä eli yrityksen asiantuntemuksen lisääntymistä ja kehittymistä, mitä voidaan hyödyntää mahdollisesti tulevissa projekteissa hyväksi ja mikä tuottaa lisäarvoa tarjontaan.

2.2 Kehitysprosessi

Projektinhallinnalla yleisesti tarkoitetaan, että työ tulisi saada tehtyä rikkomatta projektille asetettuja rajoja, esimerkiksi käytettävissä oleva aika, henkilöt ja raha. Pääasiallisina työkaluina projektinhallinnassa toimivat projektinhallintamallit, joiden avulla olisi tarkoitus saada visuaalinen kokonaiskuva projektista helpottamaan tiimin työskentelyä. Jos mallia ei olisi, se tarkoittaisi, ettei kenelläkään ole todellista kuvaa projektista. [7.]

Projektinhallinta jakautuu muun muassa projektin suunnitteluun, käynnistykseen, toteuttamisen seurantaan, ohjaukseen ja projektin päättämiseen. Suunnittelun tavoitteena on asettaa tavoitteet ja reunaehdot. Selkeä tavoite on esimerkiksi määrittelyssä *y* spesifioidun järjestelmän *x* toteutus. Viimeistään projektisuunnitelmassa tulee tarkentaa tavoitteet sellaiseen muotoon, että niiden loppuarviointi on mahdollista projektissa. Näiden lisäksi suunnitteluun pitäisi sisältyä organisointi, tavoitteiden tarkentaminen, riskien analysointi, teknologioiden ja työmenetelmien valinta, tukitoiminnot, kuten dokumentointi, tuotteenhallinta ja laadunvarmistus ja projektin osittaminen. Ongelmallisinta suunnittelussa monesti onkin projektin osittaminen ja osien aikatauluttaminen. [7.]

Pieni, mutta erittäin tärkeä osa projektin kulkua on juuri projektin käynnistäminen, sillä projektit saattavat käynnistyä vaivihkaa niin, että edes kaikki osalliset projektiryhmässä eivät ole siitä kovinkaan tietoisia. Projektiryhmällä tarkoitetaan projektipäällikköä ja kehittävää tiimiä. Projektin käynnistämisessä olisikin hyvä pitää osallistujille käynnistämistilaisuus, että kaikille osallisille ei jäisi epäselvyyttä, miten projektin tulisi edetä, ja roolit ovat selkeät. Esiteltävät asiat ovat projektin tavoitteet, tärkeys, osallistujat, sidosryhmät, aikataulu ja muut projektiin liittyvät käytännöt. Tärkeän tästä vaiheesta tekee mielestäni se, että liian usein projektit miehitetään niin sanotusti vaivihkaa projektin edetessä. [7.]

Toinen tapa, josta olisi hyvä pitää kiinni, on projektin päättymisestä viestiminen selkeästi. Se voisi olla esimerkiksi vastaavanlainen palaveri kuin käynnistäminen, mutta tässä tapauksessa tiedotetaan sidosryhmille projektin päättämisestä ja käydään läpi, miten projekti on eri vaiheissa edennyt, millaisia haasteita on tullut eteen ja mitä näille haasteille voisi jatkossa tehdä, ettei vastaavien ongelmien kanssa tarvitse käyttää ylimääräistä energiaa solmujen selvittämiseen. Tämän palaverin sisältä saadaan myös tehtyä loppuraportti muun muassa tulevien projektien suunnittelun avuksi. Projektin päättämisen yhteydessä dokumentaatio ja tiedostot siivotaan eli tarpeeton poistetaan ja tarpeellinen arkistoidaan. [7.]

Projektimalleja on monia erilaisia. Mallit erottaa toisistaan erilaiset työvaiheet ja erilainen lähestymistapa projektin kulussa. Kuitenkin päälinjojen tulee täsmätä, eli projektin aloitus, suunnittelu, toteutus ja lopetus, ja jokaiseen aiempaan vaiheeseen olisi hyvä pystyä palaamaan. Projektin kuuluu olla läpinäkyvä niin asiakkaalle kuin tiimillekin, jotta ongelmiin pystytään puuttumaan ajoissa, sillä pieni virhe ohjelmistotuotannossa saattaa kasvaa erittäin suureksi lisätyöksi myöhemmässä vaiheessa. Monet projektit ovatkin epäonnistuneet juuri siitä syystä, että havaitaan virhe liian myöhään tai pahimmassa tapauksessa salaillaan sitä.

Projektilla on pitää olla projektisuunnitelma, jossa kuvataan, miten määritetyillä resursseilla päästään tietyn aikataulun puitteissa haluttuun lopputulokseen. Tämän lisäksi suunnitelmassa kuvataan riskit, tukitoiminnot ja toteutusvälineitä ynnä muuta sellaista. Suunnitelmaa voisi pitää projektinseurannan välineenä, jonka avulla mahdolliset haasteet aikataulussa tai resursseissa havaitaan ja reagoidaan mahdollisimman ajoissa. [7.]

Esimerkki miten kehitysprosessi voisi edetä:

1. Asiakas tekee tarjouspyynnön tuotteesta tai palvelusta omien tarpeiden mukaan, ja sen perusteella tuotetta kehittävä yritys tekee esitutkimustyön.

2. Kun esitutkimus on saatu tehtyä, saadaan aikaan alustava määrittely, joissakin tapauksissa myös kevyt prototyyppi tuotteesta ilman suurempia toiminnallisuuksia.

3. Saadaan aikaan alustava määrittelydokumentti ja tarjous. Dokumentti sisältää tietoa käytettävästä laitteistosta, tarjottavista sovelluksista, asiakaskohtaisia muutoksia, puuttuvat piirteet, ulkoiset yhteydet ja alustavan projektisuunnitelman.

4. Asiakas arvioi dokumentin ja tarjouksen. Jos olemassa olevasta tarjouksesta puuttuu olennaisia osa-alueita tai asiakas kokee tarvetta muutoksille, muokataan tarjousta niiden perusteella, kunnes asiakas hyväksyy tarjouksen ja tehdään esisopimus.

5. Päästään projektisuunnitteluun, joka sisältää aikataulun, resurssit, seurannan, laatusuunnitelman, koulutussuunnitelman, asennussuunnitelman ja käyttöönottosuunnitelman.

6. Asiakas katsoo suunnitelman läpi, sitä hiotaan, jos tarvetta, ja hyväksyy, jos kaikki näyttää hyvältä ja tehdään sopimus.

7. Tämän jälkeen alkaa varsinainen toteutus. Projektisuunnitelmassa pitäisi olla määritelty malli, miten tuotetta tai palvelua aletaan kehittää.

8. Kun tuote on valmis, tuote tai palvelu arvioidaan, koulutetaan asiakas, jos tarvetta, otetaan tuote tai palvelu käyttöön ja jälkiarvioidaan niin asiakkaan kuin kehittävän tiimin kanssa.

9. Tuotteen tai palvelun luonne saattaa vaatia, että projektia jatketaan ylläpitoprojektina kestoltaan määräaikaisena tai toistaiseksi voimassa olevana.

2.2.1 Scrum-malli

Scrum antaa sovelluskehitykseen mallin, jonka avulla projektin ohjaus tapahtuu. Se ei kuitenkaan ota kantaa insinöörikäytäntöihin vaan enemmänkin keskittyy projektin vaiheistukseen ja kontrolliin, jotta projekti sujuisi mahdollisimman sujuvasti. Tätä pidetään usein hyvinkin ketteränä tapana johtaa projekteja. [8.]

Scrumista tekee erityisen ketterän se, että projektissa esiintyy vain kolme eri roolia: tuotteen omistaja, Scrum-mestari ja tiimi. Esimerkiksi vesiputousmallissa on useampia eri rooleja, mutta Scrumissa yhdellä henkilöllä voi olla useampia vastaavia rooleja ja kussakin roolissa voi olla useampia henkilöitä. Poikkeuksena tässä tapauksessa on projektipäällikön rooli. [8.]

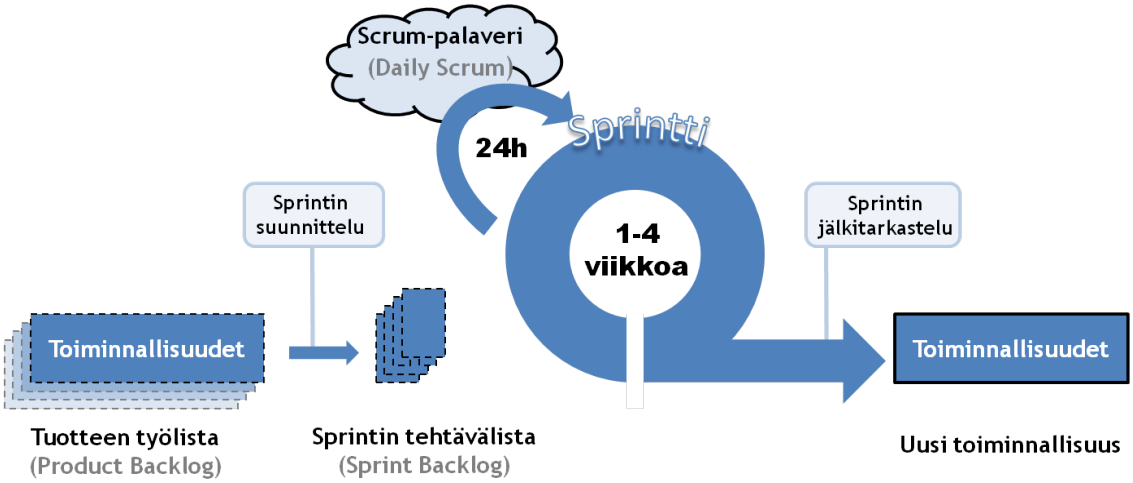
Tuotteen omistajalla tarkoitetaan henkilöä, joka vastaa tuotteen ominaisuuksista. Asiakasprojekteissa tuotteen omistaja voi olla asiakkaan oma edustaja tai toimittajan tekninen projektipäällikkö, joka tiimiä vetää. Optimitilanteessa edustaja olisi juuri asiakkaan oma edustaja, koska silloin todellinen kontrolli ja vastuu pysyvät asiakkaan puolella. Omistajan tehtävänä on saada aikaan kaikki päätökset toiminnallisuuksista ja ominaisuuksista sekä jatkokehityssuunnitelmista. Käytännössä Scrum-mallin tavoit-

teena on, että projektilla olisi vain yhdet kasvot asiakkaan suuntaan ja siten pysyttäisiin asiakkaan näkökulmasta mahdollisimman yksinkertaisella tasolla. [8.]

Scrum-mestarilla tarkoitetaan henkilöä, joka huolehtii tiimin resursoinnista ja katsoo, että tiimi toimii saumattomasti. Tiimiläiset raportoivat mahdolliset ongelmat Scrum-mestarille, ja mestarin tehtäväksi jää katsoa, että ongelmat tulevat korjatuksi. Tämän lisäksi Scrum-mestarin tehtäviin kuuluu pitää päivittäiset Scrum-palaverit ja vastata, että työt suoritetaan Scrum-menetelmän edellyttämällä tavalla. [8.]

Tiiminä toimivat kaikki henkilöt, jotka osallistuvat projektiin. Tiimiin ei erikseen nimetä sisäisiä rooleja, vaikka tiimillä olisikin omia erikoisosaamisiaan. Tiimiin kootaan tarvittava määrä henkilöitä, joilla on eri osaamisia, jotta projekti toteutuu. Tämän suurin syy on se, että Scrum-mallissa halutaan nostaa jokainen erillinen jäsen yhtä korkealle tasolle eikä ketään nosteta erikseen. [8.]

Käytännössä Scrumissa käytetään neljää dokumenttia kuvaamaan projektin etenemistä. Kuvassa 3 näytetään esimerkin kautta, miten Scrum toimii. **Tuotteen työlista** eli **Product Backlog** on tuotteen omistajan priorisoima lista, johon sisällytetään kaikki, mitä saatetaan kehittää. Listaa tulisi päivittää ja tarkentaa jatkuvasti projektin edetessä. **Julkaisun edistymiskäyrän** eli **Release Burn-up/Burn-down** avulla visualisoidaan jäljellä olevaa kehitysjonoa suhteessa julkaisusuunnitelmaan. **Sprintin tehtävälistan** eli **Sprint Backlogin** avulla kehitystiimi pystyy hallitsemaan töitään sprintin aikana. Se on ajantasainen suunnitelma tarvittavista tehtävistä ja töistä projektissa. **Sprintin edistymiskäyrässä** tarkastellaan yksittäistä sprinttiä, että työmäärä on oikeassa suhteessa aikaan. [9.]



Kuva 3. Scrum-malli [9].

Scrumin iteraatiota kutsutaan sprintiksi. Käytännössä sprintti on kiinteä ajanjakso, jonka aikana pitäisi saada aikaan julkaisukelpoinen tuote jokaisen sprintin jälkeen. Sprintti ei voi venyä pidemmäksi kuin määritelty ajanjakso, vaan sen tulisi olla tarpeeksi hyvin suunniteltu, että se toimisi oikein. Yhteen sprinttiin kuuluu Scrumin tapahtumia, kuten **sprintin suunnittelupalaveri** eli **Sprint Planning,** joka tarkoittaa nimensä mukaisesti sprintin aloituspalaveria, johon kuuluu kaksi vaihetta. Ensimmäisessä vaiheessa tulisi valita sprintin kohde perustuen tuotteen kehitysjonoon ja tiimin kykyyn tehdä niistä toimiva ohjelmisto. Tämä palaveri on myös tilaisuus, jossa tiimi voi kysellä tarkennuksia valittuihin kehitysjonon tehtäviin. Toisessa vaiheessa tulisi selvittää keinot implementoida valitut kehitysjonon tuotteet potentiaalisesti julkaistavaan tuotteeseen muun muassa jakamalla kehitysjonon asiat pienempiin tehtäviin. [10.]

Päiväpalaveri eli **Daily Scrum** on lyhyt noin 15 minuuttia kestävä palaveri joka päivä. Tämä on paikka, jossa synkronoidaan ja koordinoidaan päivittäiset tehtävät. Tarkoituksena on saada läpinäkyvyyttä tiimin sisällä: esimerkiksi jos jollakin tiimin jäsenellä on haasteita jonkin tehtävän kanssa, mitä sille voitaisiin tehdä, voisiko joku auttaa, onko jollakin tyhjempi työlista ynnä muuta sellaista. Tämä on myös paikka, jossa voidaan huomata, että kehitysjonoon valitut asiat eivät kaikki tule toteutumaan, joten voidaan ilmoittaa tuotteen omistajalle, joka päättää, mitä asialle voitaisiin tehdä. [10.]

Sprintin katselmointi eli **Sprint Review** on palaveri, jossa tarkastetaan ja arvioidaan, mitä sprintin aikana saatiin aikaan ja missä vaiheessa tuotteen kehityskaarta

ollaan. Aiheina voivat olla onnistumiset tai epäonnistumiset sprintissä. Tiimi voi tarvittaessa myös esitellä aikaansaannokset. Tämä on palaveri, johon saatetaan kutsua muita sidosryhmiä Scrum-tiimin ulkopuolelta. [10.]

Retrospektiivi- eli **Retrospective**-palaverin tarkoitus on tarkastella, miten projekti tuli tehtyä. Retrospektiivi-palaveri on tiimin kehittämiseen tarkoitettu palaveri, jossa voidaan ottaa kantaa tulevaisuuden työtapoihin, muuttaa jokin aiempi päätös tai vaikka päättää ottaa jokin uusi työkalu käyttöön tulevaisuutta varten. [10.]

Monesti Scrum-mallin haasteeksi tulee se, että rooleja saatetaan sekoittaa mallin vastaisesti. Olen nähnyt tilanteita, joissa henkilö on toiminut useassa roolissa samaan aikaan. Yleisin taitanee olla Scrum-mestarin ja tiimiläisen yhdistelmä, joskus jopa jonkinlaisella rotaatiolla eli vaihtelevalla Scrum-mestarilla. Scrum-mallissa kuitenkin on tärkeää, että tiimi saa luottamusta ja se kehittyy yhdessä itseohjautuvaksi. Scrum-mestari toimii tuotteen kehityksen ”mahdollistajana” eli paneutuu ongelmakohtiin, valmentaa tiimiä ja tuotteen omistajaa, jotta tuotteen kehitys kulkee oikeilla urilla. Toimiminen samanaikaisesti tiimin jäsenenä ja Scrum-mestarina luo ja ratkoo mahdollisia ongelmia. Usein tarvitaan kuitenkin tiimin ulkopuolinen näkemys, jotta ongelmat saadaan ratkaistua, koska tuotteen omistajalla tuskin on aikaa tähän.

Omasta mielestäni juuri Scrum on mallina paras siitä syystä, että jos tiimi saadaan nivottua tiiviiseen yhteistyöhön ja se kehittyy yhdessä eteenpäin, saadaan aikaan vahva ryhmä, johon voi aina luottaa. Scrum toimii parhaimmillaan, jos henkilöt Scrumin sisällä eivät vaihdu kovin usein. Näin ei kuitenkaan aina ole, vaan juuri rotaatioita olen nähnyt, eli sitä, että Scrum-tiimiläiset vaihtuvat esimerkiksi kahden sprintin jälkeen, mikä omalla tavallaan aiheuttaa haasteita Scrum-menetelmän käyttökelpoisuuteen.

Scrumista puhutaan ketteränä projektityömallina, ja se sopii siksi parhaiten sellaisiin projekteihin, joissa on tiukka aikataulu, jotka ovat monimutkaisia toteuttaa tai jotka ovat ainutkertaisia. Ketterä toimintatapa sopii sellaisiin projekteihin, joissa tehdään jotain täysin uutta, tai ainakin jotakin, mikä on tiimille uutta. Jos tehdään tuotetta tai palvelua, joka on tiimille jo ennestään tuttu, ketterää menetelmää ei välttämättä tarvita. [11.]

2.2.2 Kanban-menetelmä

Kanban on Toyota-yrityksen 1940-luvulla kehittämä tapa tehdä töitä. Yksinkertaistaen ajatus lähtee kahden kaistan käytössä olevasta liikenteestä. Jos toinen kaista on täysi ja toinen tyhjä, todennäköisimmin päästään nopeimmin perille tyhjää kaistaa pitkin. Tähän perustuu myös Kanbanin periaate, eli jokaisella tehtävällä on niin sanottu oma kaista, joka ei saa koskaan olla liian täysi estämään tehtävän suorittamista ilman hidasteita, vaan tehtävien suorittamisen tulisi aina sujua mutkattomasti. Periaatteessa optimitilanne on sellainen, että jokaisella kaistalla on vain yksi tehtävä, joka tehdään kerrallaan alusta loppuun ja sen jälkeen vasta aloitetaan seuraavan tehtävän tekeminen.

Kanban-menetelmä sisältää vain kolme sääntöä: **Näkyvöitä työnkulku,** mikä tarkoittaa sitä, että työt tulisi pilkkoa sopivan kokoisiin tehtäviin. Tämän lisäksi tehtävät tulisi kirjata paperilapuille ja kiinnittää Kanban-tauluun. Jokainen sarake kuvaa, missä työvaiheessa kukin tehtävä on. **Määritä Work In Progress** -taulun jokaiselle sarakkeelle tarkoittaa suurinta määrää tehtäviä, joita kyseisessä sarakkeessa saa olla. Tällä pyritään estämään, että kasaantuva työ aiheuttaisi häiriötä muille työvaiheille. **Kirjaa tehtävien läpimenoajat,** eli pitäisi tietää keskimääräinen suoritusaika yhden tehtävän valmistumiseen. Kun tämä on tehty, prosessia voidaan optimoida ja läpimenoaikaa ja ennustettavuutta voidaan parantaa. [12.]

Suurin ero Scrumiin verrattuna on siinä, ettei Kanban sisällä sprinttejä, vaan suunnittelu- ja määrittelypalaverit pidetään tarvittaessa ja tuotteen julkaisu tapahtuu heti kun työ on valmis. Tämä tarkoittaa, että Kanban-taulu ei tyhjene vaan ainoastaan pysähtyy lomien ja viikonloppujen ajaksi. [12.]

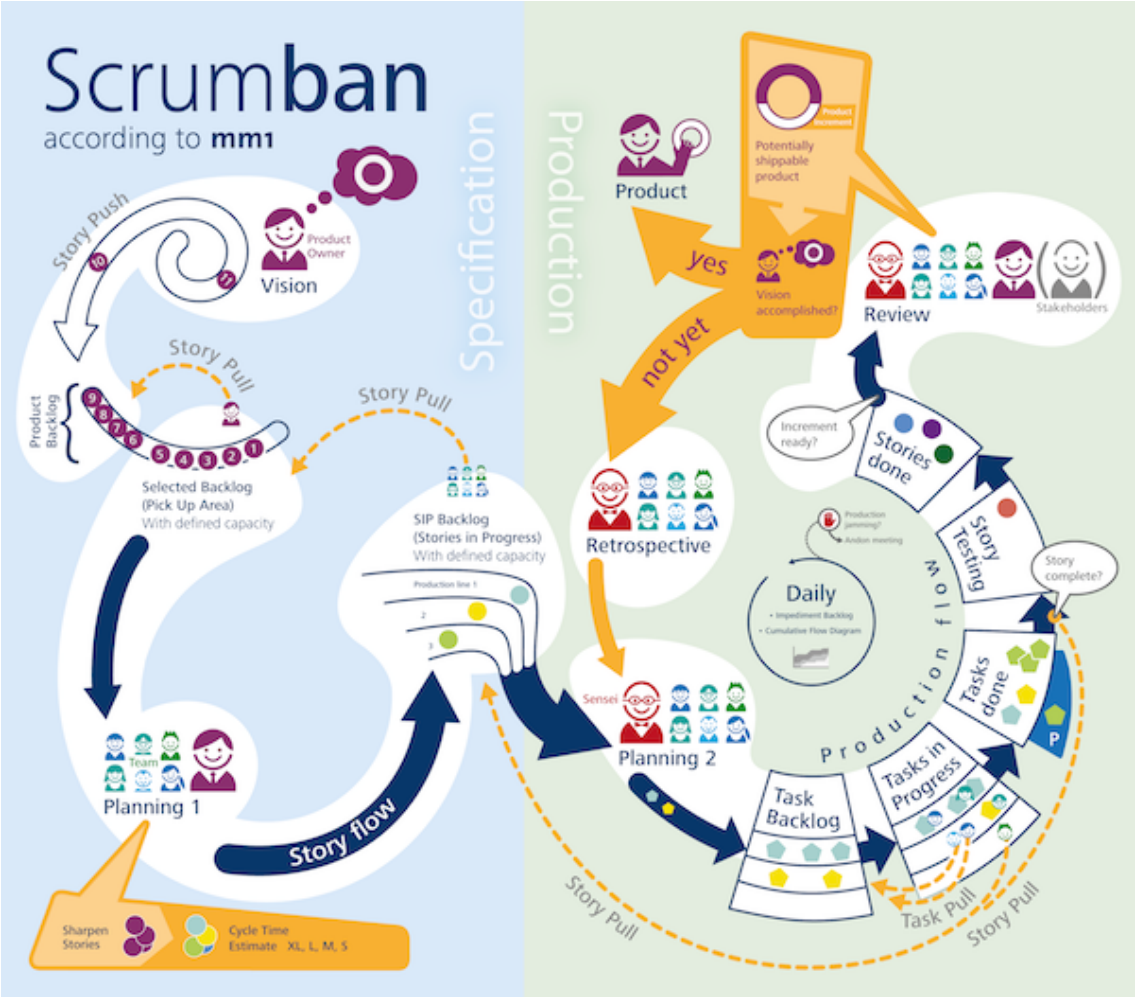
Kanban onkin erinomainen tapa tehdä projektia, kun työn sisältöä ei voida ennustaa edes viikoksi eteenpäin. Kanban usein toimii Scrumin tukena, sillä silloin pystytään reagoimaan virheraportteihin välittömästi, jos asiakas kokee, että se on tarpeellista projektin kannalta. Riskinä tässä on myös se, että voidaan menettää tavoitteellisuus, kun ei toimitakaan niin sanotusti aikaa vastaan vaan välittömästi.

Kanbania pidetään myös ketteränä projektityömallina, ja se toimii hyvin maintenance-, eli ylläpitoprojekteissa. Scrumissa on suunnitelmallisuutta ja kehitysjono, kun taas Kanban ei niitä tarvitse, vaan asioihin reagoidaan siinä hetkessä. Kanban saattaa olla

oikea valinta, jos tehdään useampia projekteja samanaikaisesti. Kuvassa 4 näytetään malli, miltä Kanban-taulu voisi näyttää.

Kuva 4. Kanban-taulu [10].

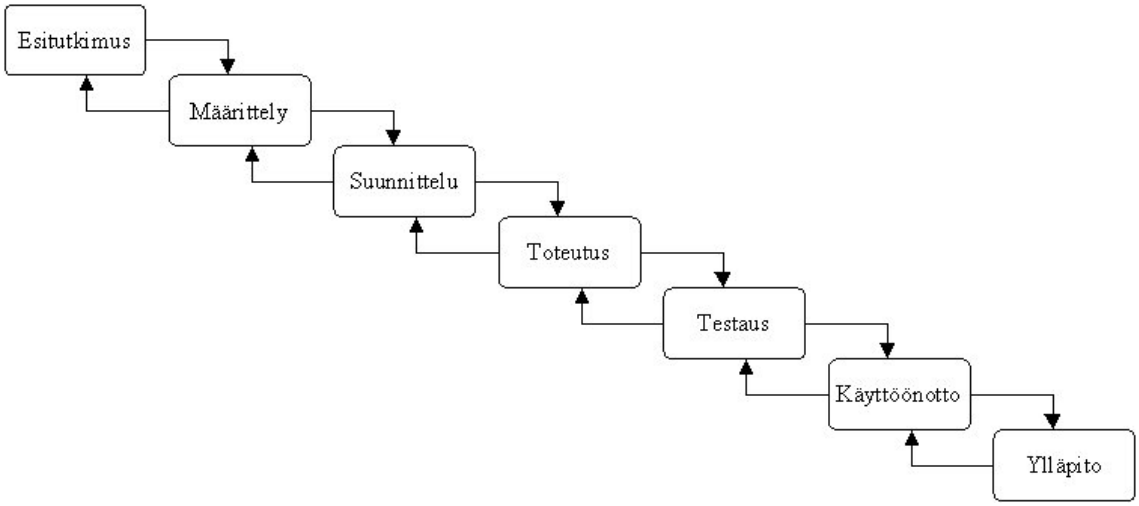
Scrumin ja Kanbanin yhdistelmää kutsutaan Scrumbaniksi, joka on suhteellisen uusi tapa toimia. Kyseessä on ketterä projektinhallintamalli, jossa toimitaan sekä sprinteissä, että Kanban-taulun tavoin. Tämä toimii hyvin, jos kyseessä on ylläpitoprojekti tai projekti, jossa tulee toistuvasti uusia vaatimuksia tai yllättäviä ohjelmistovirheitä. Itselläni on kokemusta juuri tästä mallista, jossa toimitaan rotaatiolla eli kehitystiimi työskentelee vuorottain Scrumin ja Kanbanin mukaisesti. Tämä on yrityksessämme yksi tapa motivaation ylläpitämiseen, että tekijät saavat vaihtelua päivärutiiniinsa. Tässä itse olen tosin kokenut haasteeksi juuri sitoutumisen. Toki voi olla mielekästä tekijälle saada vaihtelua töihinsä. Jos ei ole Scrumissa kuin harvoin, pysyykö tuotteen kehitys vaadittavalla tasolla? Aina kun on taas Scrumin puolella, täytyisi muistella, mitä tavoitteita tuotteen kehityksessä on, ja selvittää työlistan muuttuneet tarpeet. Lisäksi tiimi ei kehity Scrum-periaatteiden toivomalla tavalla. Kuvassa 5 näytetään miten Scrumban voisi toimia.



Kuva 5. Scrumban [13].

2.2.3 Vesiputousmalli

Vesiputousmalli on vaiheellinen ohjelmistotuotantomalli, joka tarkoittaa sitä, että projektin suunnittelu ja toteutus etenee suoraviivaisesti kuin vesi vesiputouksessa. Tämä myös tarkoittaa, että työ tulisi suunnitella huolellisesti, joten vesiputousmallissa on potentiaalia saada kustannussäästöjä, sillä hyvin suunnitellussa ohjelmistossa virheiden todennäköisyys pienenee. Käytännössä vesiputousmalli toimii melkein täydellisesti vain, jos kaikki on suunniteltu pitkään ja tarkasti. Voisi siis sanoa, että kyse on hyvinkin suoraviivaisesta kehityksestä. Kuvassa 6 näytetään, miten vesiputousmallissa edetään. [14.]



Kuva 6. Vesiputousmalli [15].

Hyötynä vesiputousmallissa on nimenomaan suunnittelu, sillä hyvä suunnittelu tarkoittaa, että eri vaiheet tulisi olla myös hyvin dokumentoitu. Kun jokainen vaihe on dokumentoitu kattavasti, on esimerkiksi uuden työntekijän helppo hypätä mukaan projektiin milloin tahansa, sillä edellinen vaihe tulisi olla hyvin dokumentoitu. Tämän lisäksi vesiputousmalli antaa hyvin selkeän, ymmärrettävän, opetettavan ja hallittavan menetelmän ohjelmistokehitykseen. [14.]

Vesiputousmallissa ongelmaksi saattaa koitua suunnitteluvaihe, sillä usein on vaikeaa nähdä lopputulosta, ennen kuin tuote on valmis. Itse koen ohjelmiston tai web-kehittämisen projekteiksi, jotka aina muuttuvat matkan varrella, oli kyse sitten jostain vastaan tulleista ongelmista tai muutostarpeista. Vesiputousmalli perustuu oikeastaan vain täydellisesti suunniteltuihin ja valmiiksi määriteltyihin projekteihin. [14.]

Melko usein kuitenkin projektien tekeminen on yhdistelmä jotain toista mallia, ja vesiputousmalliakin käytetään toisten mallien rinnalla. Näin projekteista ja kehittämisestä tulee dynaamisempaa ja riskittömämpää. Useimmat ohjelmistosuunnittelumallit ovat muokattu versio vesiputousmallista, sillä muutoksella, että ne antavat enemmän joustavuutta projektinhallintaan. [14.]

Vesiputousmallia pidetään hyvin perinteisenä projektityömallina, joka vaatii, että tuotteen määrittelyt ja vaatimukset tuotteelle tai palvelulle on tehty hyvin tarkkaan. Lisäksi tämä malli vaatii, että teknologia on hyvin tunnettu, tuotteella ei ole epäselviä määrittelyjä ja että saatavilla on runsaasti resursseja. Vesiputousta on hyvä käyttää, jos

projekti on lyhyt eikä asiakkaalta tarvita paljon vuorovaikutusta kehityksen etenemiseksi. Kun tuote tai palvelu on kehitetty, ongelman korjaaminen maksaa ketteriä keinoja enemmän, koska täytyisi päivittää kaikki dokumenteista logiikkaan. [16.]

On perusteltua käyttää aikaa ja rahaa vesiputousmallin käyttämiseen jos määrittelyvaatimukset on tehty tarkkaan, vaatimukset tuotteelle eivät muutu tai jos järjestelmällä on kriittisiä turvallisuuteen liittyviä ja hengenvaarallisia tekijöitä.

Kun tarkastelee uudempia malleja, voi kuitenkin nähdä aika paljon yhtäläisyyksiä perinteisen vesiputousmallin kanssa. Esimerkiksi Scrumia voidaan kuvailla sarjana toisiaan seuraavia lyhyitä vesiputouksia, vaikka malli onkin täysin eri.

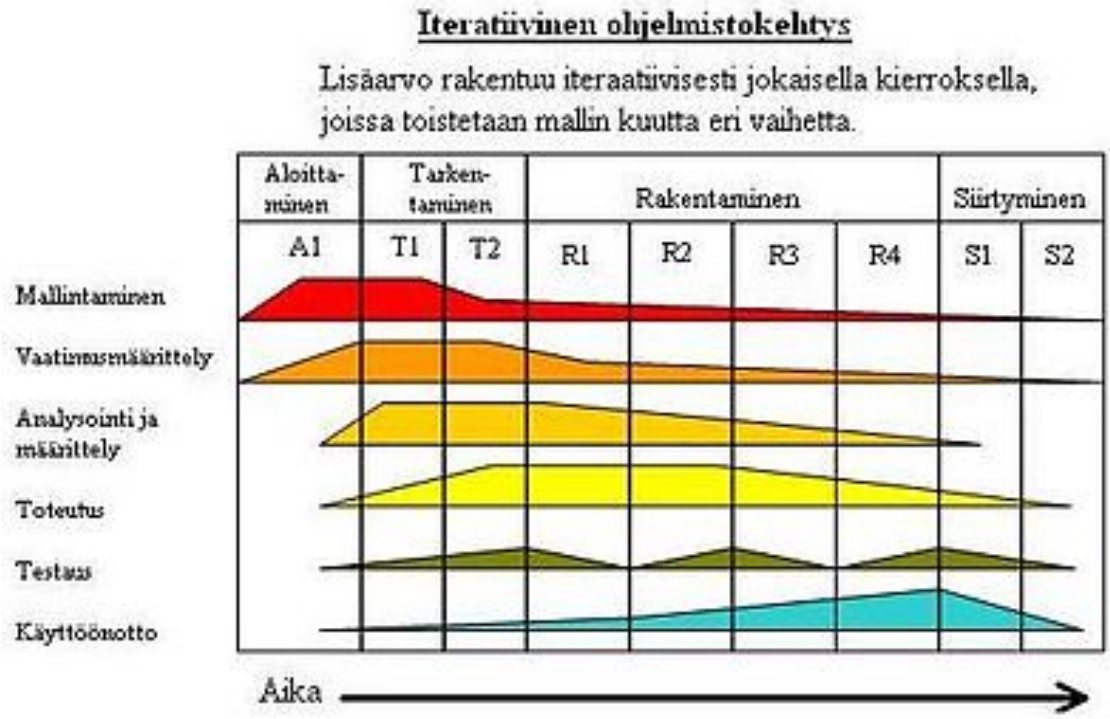
2.2.4 RUP-malli

RUP (Rational Unified Process) tarkoittaa yhtenäistettyä prosessia, joka tulee UP:sta (Unified Process), eli iteratiivisen ohjelmistokehityksen prosessimalli. Tämä malli usein räätälöidään tarpeiden mukaan, sillä kyseessä on varsin laaja projektinhallintamalli. Käytännössä RUP:ssa on neljä päävaihetta: **aloittaminen**, **tarkentaminen**, **rakentaminen** ja **siirtyminen**. Jokaisessa vaiheessa on kuitenkin yksi tai useampia iteraatioita, ja yhteen iteraatioon sisältyvät kaikki vaiheet, joita käytetään vesiputousmallissa. Erona on, että työvaiheiden painotukset vaihtelevat iteraatiosta riippuen. [17.]

Tässä mallissa ei käytetä varsinaisesti organisaatiomalleja, vaan asiat tehdään roolitusten perusteella. Rooleja ei käsitellä työnimikkeillä tai henkilöinä, vaan rojektin roolitus on projektipäällikön vastuulla. Dokumentointi vaihtelee projekteittain, mutta suosituksena on, että työt ja tuote on kuitenkin edes jollakin tasolla dokumentoitu. Koska kyseessä on monta prosessia samanaikaisesti, dokumentointi selkeyttää projektin etenemistä. [17.]

RUP:n hyödyiksi voitaisiin sanoa, että sen pitäisi vaatia vähemmän aikaa integrointeihin, kun iteraatiot tapahtuvat samanaikaisesti. Kolikon kääntöpuolella tosin tämä tapa kehittää tuotetta vaatii erittäin vahvat tekijät taustalle, sillä työ on monimutkaista ja voisi sanoa, että se on epäorganisoitua. Integroinnit läpi prosessien kuulostaa teoriassa hyvältä, mutta varsinkin isoissa kehitysprojekteissa useat iteraatiot saattavat aiheuttaa hämmennystä ja ongelmia testausvaiheessa. RUP sopii projektityömallina

suuriin ohjelmistotuotannon projekteihin. Kuvassa 7 kuvataan iteratiivisen ohjelmistokehittämisen vaiheet, ja vaiheiden samanaikaisuus.



Kuva 7. RUP-malli [14].

2.2.5 Extreme Programming -metodi

Extreme Programming on yksi ketterän kehityksen metodeista. Tämän metodin tekee hyväksi sen tehokkuus. Sen sijaan, että julkaistaan jotain tiettyyn päivään mennessä, julkaistaan vain se, mitä tarvitaan. Tämä vaatii kehittäjiltä itsevarmuutta vastata asiakkaan vaatimuksiin heti, kun niin tahdotaan. [18.]

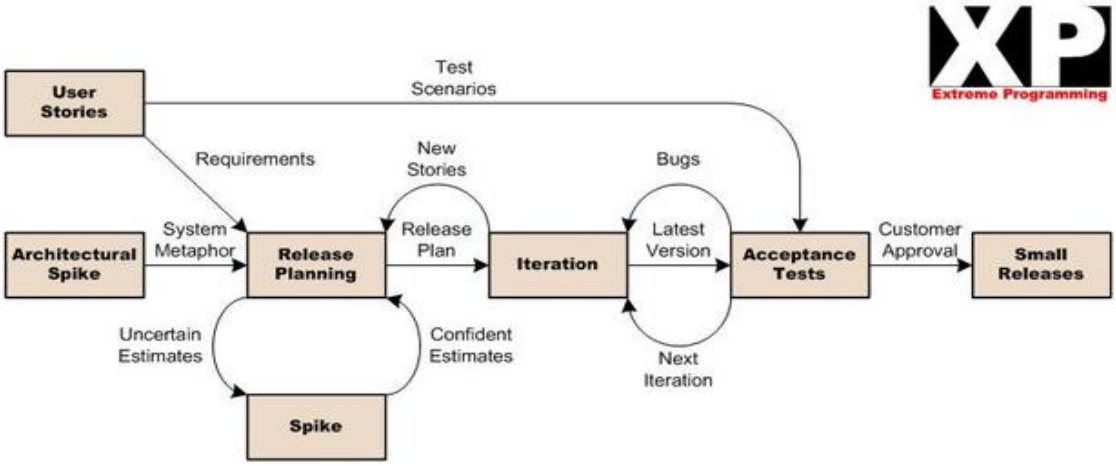
Metodi painottaa juuri tiimityötä eli sitä, että päälliköt, asiakkaat ja kehittäjät ovat kaikki samanarvoisia kumppaneita ja tekevät yhteistyötä tiimissä. Tämä myös antaa tiimille mahdollisuuden tulla erittäin tuottavaksi, itseorganisoituvaksi ryhmäksi. [18.]

Metodin tavoitteena on, että kommunikointi paranee, projekti yksinkertaistuu, palautteen saaminen on välitöntä, luottamus ja kunnioitus kasvaa ja tiimille tulee rohkeutta

kokeilla uusia asioita. Tämä onkin syy, miksi Extreme Programming on monessa yrityksessä käytössä. Itselleni metodi on hieman vieraampi tapa toimia. [18.]

Yllättävintä Extreme Programming -metodissa ovat sen periaatteessa yksinkertaiset säännöt. Ne saattavat kuulostaa monimutkaisilta: **Työn suunnittelu** tarkoittaa, että kirjoitetaan käyttäjäkertomus, suunnitellaan julkaisut ja tehdään aikataulutus, tehdään useita pieniä julkaisuja ja tehdään iteraatiosuunnitelma. **Työn hallinta** tarkoittaa, että annetaan tiimille tarvittava avoin työtila, annetaan toteutettavissa oleva tahti, pidetään päivittäinen palaveri kuten Scrumissa, mitataan työtahtia ja korjataan metodi, kun sille on tarvetta. **Tuotteen suunnittelu** tarkoittaa, että tehdään tuotteesta yksinkertainen, kerrotaan esimerkein mitä tuotteen tulee tehdä, ei lisätä toiminnallisuuksia liian aikaisin ja refaktoroidaan, kun mahdollista. **Ohjelmointi** tarkoittaa, että asiakkaan tulisi olla aina tavoitettavissa, ohjelmoinnin on tapahduttava sovituin standardein, tehdään ensin yksikkötestausta, kaikki tuotanto-ohjelmointi tehdään pari-ohjelmointina, vain yksi pari integroi kerrallaan, integroidaan usein ja käytetään vain tiettyä konetta integrointeihin. **Testaus** tarkoittaa, että kaikella koodilla on oltava yksikkötestausta, jokaisen koodin täytyy läpäistä yksikkötestaus ennen julkaisua, kun virhe löytyy, täytyy luoda testit, hyväksyntätestaus täytyy tehdä usein ja tulokset on julkaistava. Käytännössä metodissa ohjelman kehitys on jaoteltu siis moniin pieniin osiin ja osat näyttävät järkeviltä vasta, kun tuote on julkaistu. [18.]

Extreme Programming -metodi on kehitetty pieniä tai keskisuuria ryhmiä varten, kun ohjelmistoa kehitetään jatkuvasti muuttuvien vaatimusten keskellä. Käytännössä tällä metodilla on tarkoitus yhdistää hyväksi havaitut käytännöt muista malleista ja periaatteet sen taustalta, ja ne viedään äärimmäisyyksiin. Kuvassa 8 näytetään esimerkki, miten Extreme Programming -metodi toimii käytännössä.



Kuva 8. Extreme Programming [18].

3 Pohdintaa ja kokemuksia projektinhallinnasta

Kuten luvussa 1 mainitsin, olen aiemmin toiminut teknisenä asiantuntijana ja siitä siirryin projektipäälliköksi. Tämä tapahtui kutakuinkin vahingossa. Tarkoitan sitä, että vastuu työpaikkani digitaalisten palveluiden suunnitteluun, kehitykseen ja ylläpitoon tuli sijaisuuden seurauksena. Sitä ennen olin toiminut ainoastaan teknisissä ylläpito- ja asennustehtävissä sekä erikoisratkaisujen mahdollistamisessa. Digitaaliset palvelut tarkoittavat tässä tapauksessa yritykseni verkkosivujen ja mobiili-applikaation suunnittelua ja kehitystä nykyisen työpaikkani yhteistyökumppanina.

Olen toiminut asiakkaan ja tuotetta kehittävän yrityksen roolissa. Tämän lisäksi olen myös kuulunut Radiot.fi-palvelun ohjausryhmään, eli voisi sanoa, että olen saanut ainutlaatuisen tilaisuuden toimia monessa eri roolissa, mitä tulee projektien läpivientiin.

3.1 Roolini asiakkaana

Asiakkaan roolista ohjelmistotuotannon kokemukseni saa alkunsa ja samalla vedettyä mukaan mielenkiintoiseen maailmaansa. Aiemmin olin ollut teknisenä asiantuntijana ja mukana pienessä sivuroolissa ohjelmistotuotannossa. Rooli kuitenkin vahvistui vuosi vuodelta, ja lopulta olinkin melkein täysipäiväisesti projektien parissa.

Mielenkiintoista tässä oli asenteeni kehittävää yritystä kohtaan. Silloin ajattelin, että sen tulisi tehdä töitä juuri silloin, kun haluan. En ymmärtänyt, että meidän kanssamme töitä tekevä tiimi on osa monia muita projekteja, joten työmäärät ja aikataulut tulisi suunnitella tarkkaan. Mielessäni oli vain ajatus, että kun maksamme x määrän rahaa tuotteen kehityksestä, meidän tulisi saada työtä sitä rahaa vastaan heti.

Tämä oli aikaa, jolloin en ymmärtänyt projektityöskentelyn tavoista, mutta kokemuksena äärimmäisen tärkeä vaihe urallani. Jos joku olisi silloin minulle sanonut, että tulen tekemään projektipäällikkönä vastaavia töitä, en todennäköisesti olisi uskonut, sillä en nähnyt itseäni muussa kuin asiantuntijan roolissa.

3.2 Roolini ohjausryhmässä

Ohjelmistotuotannon seuraava askel minulle oli olla osa Radiot.fi-palvelun ohjausryhmää, eli olimme osana kilpailuttamassa tarjouksia yhden vision taustalle. Visio oli tuottaa yhteinen digitaalinen radiopalvelu, josta kaikki radiot olisivat kuunneltavissa yhden kanavan kautta.

Ohjausryhmän tarkoituksena on antaa mahdollisimman laaja-alainen asiantuntemus projektille. Sen keskeinen tehtävä on seurata, että hanke toteutetaan projektisuunnitelman ja rahoituksen mukaisesti. Käytännössä tällä tarkoitetaan, että kerättiin ryhmä eri radioiden edustajia mahdollistamaan mahdollisimman hyvä lopputulos. Tavoitteena oli, että radioalan ammattilaiset ja kuulijat olisivat tyytyväisiä lopputulokseen.

Tämä oli minulle ensimmäinen projekti, jossa olin mukana ennen tuotteen olemassaoloa. Sain olla mukana vaikuttamassa siihen, mihin suuntaan palvelua aiotaan viedä. Ensimmäisenä ohjausryhmän tavoite oli käydä läpi, mitä halutaan saavuttaa, ja otettiin esille mahdollisia olemassa olevia tekniikoita ja työkaluja sen mahdollistamiseen. Päätimme, että tämä palvelu vaatii täysin uuden alustan, jotta tuote olisi mahdollisimman hyödyllinen loppukäyttäjälle. Tämän jälkeen tehtiin kilpailutus eri kehitysyritysten kesken, ja saimme tarjouksia ja prototyyppejä. Niiden perusteella ja usean kierroksen kautta valittiin yhteistyökumppani, joka palvelun lopulta tuotti ohjausryhmän vision mukaisesti.

Tämä oli mielestäni myös yksi palkitsevimmista projekteista, joissa olen ollut mukana vaikuttamassa. Syy siihen on se, että tehtiin jotain aivan uutta tyhjästä ja onnistuttiin projektissa mielestäni tehokkaasti ja tuotteesta tuli hyvä.

3.3 Roolini projektipäällikkönä

Projektipäällikkönä olen toiminut nyt noin vuoden verran. Tässä roolissa vastaan viime kädessä siitä, että tuote tai palvelu on oikeanlainen, valmis aikataulussaan ja pysynyt sovitussa budjetissa. Tämä tarkoittaa myös sitä, että tulisi olla hyvä kommunikoija niin asiakkaan kuin tiimin suuntaan, jotta saadaan mahdollisimman hyvä yhteistyö aikaiseksi.

Projektipäällikön tulisi myös viestiä jatkuvasti projektin tapahtumista ja olla tietoinen projektien tilanteista, seuraavista toimenpiteistä ja niiden tärkeysjärjestyksestä. Lisäksi projektipäällikön tehtäviin kuuluu erilaisten ongelmien ratkonta mahdollisimman nopeasti.

Leikkimielisesti voisi siis sanoa, että kyseessä on eräänlainen asiakkaan ja tiimin ”paijaaja”, eli pidetään myös huolta projektin sidosryhmistä ja tiimistä parhaalla mahdollisella tavalla. Hyviä ominaisuuksia projektipäällikölle ohjelmistotuotannossa ovat myös tekninen osaaminen, asiakkaan liiketoiminnan ja alan tunteminen, liikkeenjohdon ja laskentatoimen tunteminen sekä sopimusteknisten asioiden hallinta.

Tämä rooli on istunut minulle oikein hyvin, ja pidän työstäni kovasti. Hyppäsin mukaan olemassa olevaan projektiin digimarkkinoinnin parissa, ja tavoitteeni on puristaa olemassa olevasta digimarkkinoinnista kaikki hyöty. Kehitystä on tapahtunut muun muassa tehostamalla prosesseja ja käytäntöjä esimerkiksi a/b -testauksen avulla.

4 Yhteenveto

Ei ole yhtä oikeaa ratkaisua projektimallin valitsemiseen, sillä monissa on omat hyötynsä ja haittansa. Malli tulisikin valita projektin luonteen mukaan eikä sen mukaan, mihin suuntaan on ”uskovainen” eli mistä tavasta tehdä projekteja pitää eniten. Koen itse, että tärkeintä projektityössä on, että myydään mieluummin asiantuntemusta kuin projektia. Tällä tarkoitan, että tiimi pitäisi nostaa vahvemmin esille. Tämä toimii mielestäni myyntivalttina, sillä uskon, että asiantuntemus merkitsee paljon.

Mielenkiintoisia yhtäläisyyksiä malleista tosin löytyi. Puhutaan selkeästi samoista projektivaiheista, mutta niitä käsitellään ja kutsutaan hieman eri tavalla. Oma ajatukseni näiden projektimallien syntyperästä on oikeastaan tullutkin siitä, että on lähdetty tekemään jollakin aiemmin tutulla mallilla ja se ei olekaan aivan istunut omaan tekemiseen. Uskon myös, että tulevaisuudessa tulee olemaan lisää erilaisia tapoja vetää projekteja läpi ja keksitään taas uusi ”totuus”, jonka mukaan projektit tulisi tehdä. Mutta kaikki kuitenkin juontaa juurensa perusajattelusta projektien läpiviennissä.

Tämä insinöörityön tekeminen avasi ajatuksia projektityöstä ja siitä, miten haluan itse jatkossa vetää projekteja. Menen jatkossa syvemmälle aiheeseen, sillä tavoitteenani on tulla sekä asiakkaan että työpaikkani näkökulmasta vahvaksi projektin vetäjäksi.

Lähteet

1 Jokainen projektisuunnitelma on kolmio. 2007. Verkkodokumentti. Microsoft. https://support.office.com/fi-fi/article/Jokainen-projektisuunnitelma-on-kolmio-2b74c21b-a406-4727-8d74-26648a56924a. Luettu 25.9.2016.

2 Stanleigh, Michael. Risk Management…the What, Why, and How. Verkkodokumentti. https://bia.ca/risk-management-the-what-why-and-how/. Luettu 30.3.2017.

3 Storås, Niclas. 2014. It-hanke epäonnistui täysin – Tiedolla oli ”sisäisiä haasteita ja kommunikointi ongelma”. Verkkodokumentti. http://www.tivi.fi/Uutiset/2014-03-09/It-hanke-ep%C3%A4onnistui-t%C3%A4ysin---Tiedolla-oli-sis%C3%A4isi%C3%A4-haasteita-ja-kommunikaatio-ongelma-3208580.html. Luettu 30.3.2017.

4 Projektin johdon pätevyys 3.0. Verkkodokumentti. PRY. http://www.pry.fi/files/108/PMAF_NCB_3.0_v1.3.pdf. Luettu 30.3.2017.

5 Kähönen, Päivi. 2016. Projektitiimin jäsenten motivointi, osa 1. Verkkodokumentti. http://www.pasaati.com/blog/projektitiimin-j%C3%A4senten-motivointi Luettu 30.3.2017

6 Tukel, Oya & Rom, Walter. 2001. An empirical investigation of project evaluation criteria. Verkkodokumentti. http://www.emeraldinsight.com/doi/pdfplus/10.1108/01443570110364704. Luettu 30.3.2017.

7 Haikala I. & Märijärvi J. 2006. Ohjelmistotuotanto. Talentum.

8 Ketteryys haltuun: Scrum pähkinänkuoressa. 2013. Verkkodokumentti. Sininen Meteoriitti. http://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-scrum-pahkinankuoressa/. Luettu 25.9.2016.

9 Koro, Juho. 2011. Scrum. Verkkodokumentti. https://hlab.ee.tut.fi/hmopetus/vpsist-oppimateriaali/4-menetelmia-ja-malleja/4-4-ketterat-menetelmat/4-4-1-scrum.html. Luettu 25.9.2016.

10 CSM-kurssimateriaali. 2016. Reaktor.

11 Cohn, Mike. 2011. Deciding What Kind of Projects are Most Suited for Agile. Verkkodokumentti. https://www.mountaingoatsoftware.com/blog/deciding-what-kind-of-projects-are-most-suited-for-agile. Luettu 31.3.2017.

12 Lekman, Lare. 2009. Mikä ihmeen Kanban? Verkkodokumentti. https://lekman.fi/2009/09/26/mika-ihmeen-kanban/. Luettu 28.3.2017.

13 Brinker, Scott. 2014. Using Scrumban (Scrum + Kanban) for agile marketing. Verkkodokumentti. http://chiefmartec.com/2014/12/using-scrumban-lean-agile-marketing/

14 Vesiputousmalli. 2016. Verkkodokumentti. Wikipedia. https://fi.wikipedia.org/wiki/Vesiputousmalli. Luettu 27.9.2016.

15 Okol. Kehittämistyön vaiheet ja elinkaarimallit. Verkkodokumentti. http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm

16 What is Waterfall model- advantages, disadvantages and when to use it? Verkkodokumentti. ISTQB Exam Certification. http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/. Luettu 31.3.2017.

17 Rouse, Margaret. 2007. Rational Unified Process (RUP). Verkkodokumentti. http://searchsoftwarequality.techtarget.com/definition/Rational-Unified-Process. Luettu 27.9.2016.

18 Wikispaces. Extreme Programming. Verkkodokumentti. https://7bsp1018.wikispaces.com/eXtreme+Programming. Luettu 27.3.2017.